

Op Console PC

OS/400

Console session flow

 $I_b = QCONSOLE$, $H(P_b) = H(QCONSOLE$ $I_{vs} = QSECOFR$, $H(P_{us}) = H(QSECOFR)$ $I_{va} = 22222222$, $H(P_{va}) = H(22222222)$ $I_{v_1} = 111111111, H(P_{v_1}) = H(111111111)$ Shipped with: 2)use PKCS-5 to encrypt P_D with P_A 1) prompt for I_D, P_D, P_A, I_{Ux}, P_{Ux} prompt for I_b , P_a , I_{Ux} , P_{Ux} Setup wizard -Normal flow -

device EKE flow with H(P_D)

derive K_D set $P_c = K_c$ if first m

set $P_D = K_D$ if first use of P_D

derive K_D set $H(P_D) = H(K_D)$

derive K₀

user EKE flow with H(P_u)

derive $\mathbf{K}_{\scriptscriptstyle \mathrm{U}}$

secure console session encrypted with K_U

NOTE: The first console session uses the well known shipped device identifier and user ID to access the iSeries. The device passphrase is modified in the initial flow $(P_D = K_D)$. Therefore, the genesis device essentially "gets in free."

Legend:

 $I_{\rm b}$ = Device identifier

 $P_D = Device$ shared secret $P_A = Access$ passphrase

 $I_{Cx} = User ID$

 $P_{ux} = User passphrase$

 K_D = Device session key K_U = User session key

K. Random number

180

- generate DH parameters g and p
- where g = base; p = prime; these values do not have to be secret (a c + ab) c + ab
 - make g and p constants in server and client EKE code

generate R and do DH Phase 1 <-- send (Phase 1 public-info)</pre> derive K from DH Phase 2 make g, p constants generate challenge B server EKE authenticate user A <-- send H(P_D)[public info], K[challenge B] K[H(challenge A, challenge B)] K[challenge A, challenge B] device ID, H(P_D)[public-info] generate R and do DH Phase 1 derive K from DH Phase 2 authenticate server B make g, p constants generate challenge A client EKE <--- send send -->

Refer to BSAFE Reference Manual for description of DH Phase 1& 2.

NOTE: The challenge strings must be a different length than the encryption block.

the state of the s

28/

Control Progran EKE_Handshake generate device R and do DH Phase 1 generate device challenge B derive Ko from DH Phase 2 Server EKE (version, key strength) **EKE** Interface EKE negotiation device ID, EKE parms, H(Po)[public-info] EKE negotiation (version, key strength) H(P_D)[public info], K_D[challenge B] generate device R and do DH Phase 1 derive K_D from DH Phase 2 EKE negotiation response **EKE** Interface Client EKE EKE_Handshake Console

Pass 1 for device complete, begin Pass 2 for user...

authenticate Console device

K_D[H(challenge A, challenge B)]

authenticate server

Ko[challenge A, challenge B]

generate device challenge A

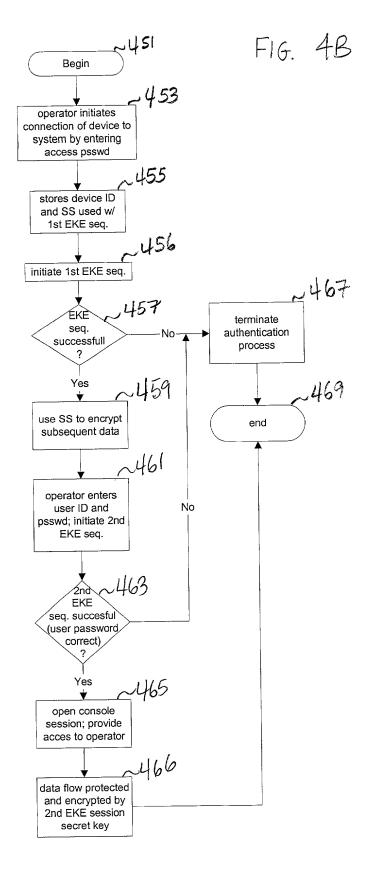
1

.4

히	Client EKE	Server EKE
Console	EKE Interface	EKE Interface Control Program
	generate user R and do DH Phase I	
	user ID, H(P _U)[public-info]	
		generate user R and do DH Phase 1 generate user challenge B derive K _U from DH Phase 2
	H(P ₀)[public info], K ₀ [challenge B]	B]
	derive K ₀ from DH Phase 2 generate user challenge A K ₀ [challenge A, challenge B]	
	K _u [H(challenge A, challenge B)]	authenticate user
	authenticate server	
EKE_Write	Ku[console data]	EKE_Read
EKE_Read	K _v [console data]	process data EKE_Write

there there were green of the first terms and the first terms to the first terms to the first terms to the first terms to the terms to

ļ.s



H(password)

H(password)

QSECOFR

H(password) H(password)

|dentifier |111111 |2222222

QSRV

password

F165A

F16 5B

fash (device dentifier shared device dentifier shared device dentifier shared decret)	Server	Device Table / Device Identifier Hashed shared secret	GCONSOLE H(shared secret) NEVICE2 Hishared	Secret	User Table
		0		0	· de la companya de l